



**HAL**  
open science

# Natural Number Game: Students' activity using an interactive theorem prover

Athina Thoma, Paola Iannone

► **To cite this version:**

Athina Thoma, Paola Iannone. Natural Number Game: Students' activity using an interactive theorem prover. Thirteenth Congress of the European Society for Research in Mathematics Education (CERME13), Alfréd Rényi Institute of Mathematics; Eötvös Loránd University of Budapest, Jul 2023, Budapest, Hungary. hal-04410448

**HAL Id: hal-04410448**

**<https://hal.science/hal-04410448>**

Submitted on 22 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Natural Number Game: Students' activity using an interactive theorem prover

Athina Thoma<sup>1</sup> and Paola Iannone<sup>2</sup>

<sup>1</sup>Southampton Education School, University of Southampton, UK; [a.thoma@soton.ac.uk](mailto:a.thoma@soton.ac.uk)

<sup>2</sup>School of Mathematics, University of Edinburgh, UK

*In this paper, we report aspects of mathematics undergraduate students' activity when writing a mathematical proof in an interactive theorem prover. The participants, two first-year undergraduate mathematics students, were invited to a series of interviews and asked to engage with a learning resource, the Natural Number Game, designed to introduce and scaffold the use of the interactive theorem prover. The students were asked to prove statements in the Natural Number Game, share their screens and use the think aloud protocol. We use the instrumental approach to explore the schemes that students develop when they write a mathematical proof using the interactive theorem prover. Our analysis illustrates the two schemes ("verify step-by-step", "verify multiple steps") that these students seem to develop while using the interactive theorem prover.*

*Keywords: Lean, interactive theorem prover, programming, proof writing.*

## Programming in university mathematics and interactive theorem provers

Modules involving programming in mathematics degrees in the United Kingdom (UK) are now more frequent than they were ten years ago (Iannone & Simpson, 2022) especially in the first and second year of mathematics undergraduate studies. Sangwin and O'Toole (2017) investigated the programming languages that are taught in mathematics departments in UK universities. Their results illustrated that the most common languages in use were MATLAB and the statistical package R, showing that programming is nearly exclusively used for applied mathematics (computational modules) and statistics modules and not pure mathematics modules. In pure mathematics and computer science research, interactive theorem provers (ITPs) have been used since de Bruijn's creation of the seminal Automath prover in the late 1960s (de Bruijn, 1980), but it is only very recently that ITPs are introduced in undergraduate teaching (Avigad, 2019; Thoma & Iannone, 2022). The use of ITPs in teaching, in the UK context at least, is frequently optional and therefore relatively small numbers of students opt to use it (Thoma & Iannone, 2022; Iannone & Thoma, 2023).

There are several ITPs currently in use in mathematics research and they differ in terms of language and interaction mode; automation and user interface; proof structure and proof state visualisation (Bartzia et al., 2022). One of the ITPs is Lean (<https://leanprover-community.github.io>). Lean, among its other features, provides instant feedback on the logical coherence of the proof and the symbolisms used. The interface is divided into two sections (Figure 1 – featuring a screenshot from the Natural Number Game and the embedded Lean environment): the coding (left) and feedback section (right). The right section, shows at the top the context and goals at the given line of code and at the bottom provides feedback regarding the code (e.g., error messages).

Avigad (2019) discusses Lean's potential impact on students' understanding while clearly considering the different languages that are involved: natural, mathematical, and programming

language. Additionally, Hanna et al., (2023) consider Lean's features which could potentially support students' difficulties with proof. Such features include the consideration of the proof strategy prior to starting to code, its feature of instant feedback, and that students' need to engage with elementary formal logic to be able to code in Lean. The results of these theoretical discussions on Lean indeed align with results from empirical studies on students' use of Lean (e.g., Thoma & Iannone, 2022).

Our previous work examined mathematics undergraduate students' pen and paper proofs by focusing on the qualitative differences of Lean and non-Lean users' proofs (Thoma & Iannone, 2022). The main differences observed were the precise introduction of the mathematical objects involved in the proof, and the often overt breakdown of the proofs in goals and subgoals. However, only a small number of students used Lean over the course of that study. Further investigation on students' perceptions of Lean confirmed students' difficulties with the syntax and illustrated that students saw coding in Lean and proving as disconnected activities (Iannone & Thoma, 2023). In this paper, we explore students' activity with the ITP and aim to contribute to the literature which investigates the link between programming and mathematics learning for undergraduate students following the call by Lockwood & Mørken (2021) stressing the importance of this under-researched area.

## **Instrumental approach**

The link between programming and learning mathematics has been explored in recent studies at university level using the instrumental approach (Gueudet et al., 2022). The key theoretical considerations of this approach are the artefacts (products of human activity designed for goal-directed activities) which together with schemes (an organization of the coder's activity for a given goal) create an instrument. A scheme comprises of four aspects (Vergnaud, 2009, p. 88):

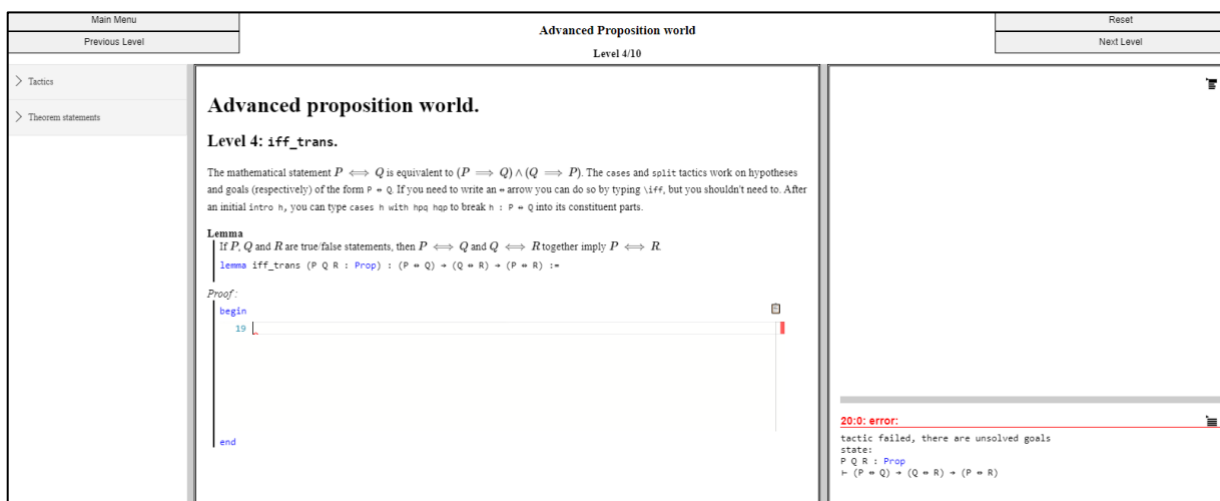
- Intentional aspect: the goal(s) of the activity;
- Generative aspect: the rules-of-action which are “the sequences of actions, information gathering, and controls” (ibid, p. 88);
- Epistemic aspect: the operational invariants which are concepts and theorems-in-action; and
- Computational aspect: the possibilities of inferences.

The same artefact (e.g., Lean) can be used for the same goal (e.g., write a proof in Lean) but with different schemes which means that users (in our case students coding in Lean) create different instruments. Furthermore, there are two aspects of the process of development of the instrument: *instrumentation* and *instrumentalisation*. Instrumentation “describes the development of knowledge by the subject: the features of the artefact influence the subject's activity and the knowledge developed” (Gueudet et al., 2014, p. 140). Whereas if “the user has specific knowledge before engaging in the interaction with an artefact and this knowledge shapes the interaction” this is called instrumentalisation (Gueudet et al., 2014, p. 140).

In our analysis, we consider students' use of the same artefact (Lean) and investigate students' development of schemes as they become apparent from their rules-of-actions when they use Lean to write the proof of the same statement. Our research question is: Which are the schemes that students develop when using Lean, in the context of the Natural Number Game, to write a mathematical proof?

## Context and methods

The data we report in this paper are part of a larger study which focuses on student's activity when using Lean. In a first-year undergraduate mathematics module offered at a UK research-intensive university, Lean was introduced by the lecturer as a voluntary part of the syllabus. The learning materials included lecture notes on the use of Lean, support materials and extra weekly tutorials on Lean. A questionnaire was administered to all students asking them to comment on their use of Lean and invited them to participate in an interview. Four students replied and were interviewed multiple times, on average 6 to 7 one hour interviews per student, after the final examination for the module. The interviews were conducted via video conferencing, and students were invited to engage with the levels of the Natural Number Game<sup>1</sup> (see Figure 1), share their screen and use the think aloud technique. In this paper, we discuss the schemes that two (Gemma and Matt) of the four students developed while writing a proof of the same statement in Lean. Our analysis focused on identifying the components of schemes (descriptions of actions and reasons for actions) in students' interviews using both audio and video data.



**Figure 1: Screenshot from the Natural Number Game Level 4 - Advanced Proposition world**

In this level of the Natural Number Game, students were asked to prove the transitive property using Lean (Figure 1). The interface contains two main sections. In the first (on the left) the user writes Lean code and interacts with the tool by altering the context. On the right, the top window displays the goal (to avoid confusion with the goals connected with the schemes we will use the phrase Lean goal to refer to this type of goals from now on) as it is altered by the user and the bottom one displays error messages. Also, there is further explanation regarding the mathematical task to be proven (in our case the transitive property) with instructions on using certain *tactics* to assist in the coding of the proof and features regarding the key symbol use (Figure 1). Furthermore, on the left-hand side there

---

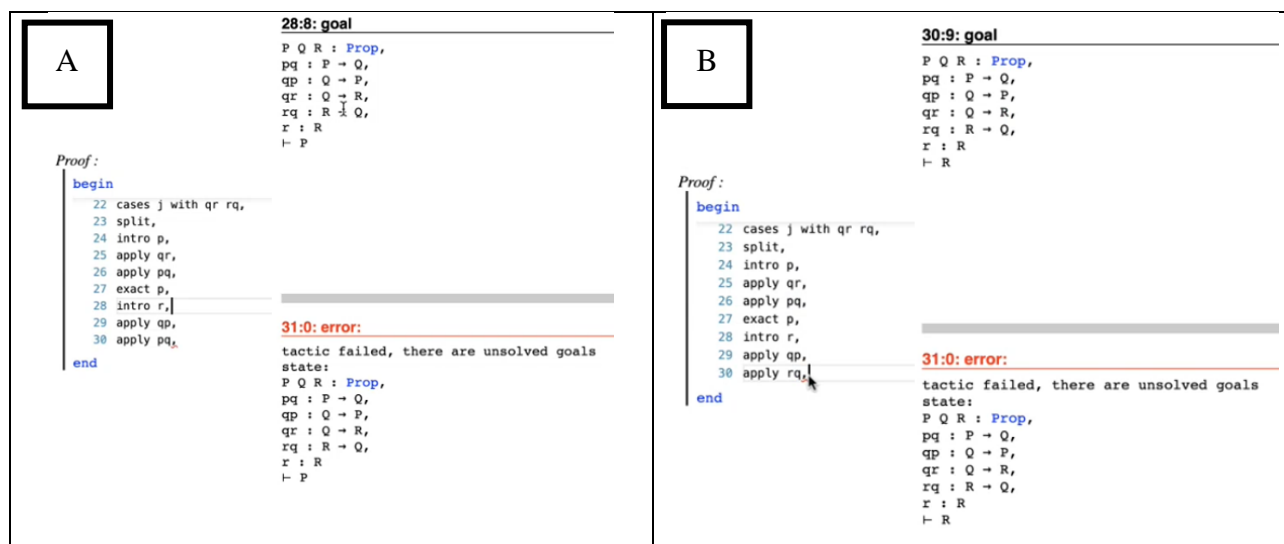
<sup>1</sup> A teaching resource created by Kevin Buzzard and Mohammad Pedramfar at [https://www.ma.imperial.ac.uk/~buzzard/xena/natural\\_number\\_game/](https://www.ma.imperial.ac.uk/~buzzard/xena/natural_number_game/)

is a drop-down menu which contains the key *Tactics* and *Theorem Statements* that have been used so far in the game.

## Gemma's and Matt's proofs of the transitive property

Gemma and Matt both worked on the level in Figure 1 during the 3<sup>rd</sup> and 6<sup>th</sup> interviews respectively. At the time of the interview, they had already successfully completed all previous levels of the game and by the end of the interview cycle Matt engaged with all 10 worlds of the game while Gemma with 7. We chose to discuss Matt and Gemma's proofs (Figures 4 feature the complete proofs) as these illustrate the development of different schemes.

Gemma started by using *intro h* aiming to check what she has to deal with first. We interpret this as a rule-of-action "explore what is given". Then, she used initially *cases h with p q* which, after checking that it was making the relevant changes in the context of her proof, she then improved to *cases h with pq and qp* aiming to provide further clarity in the symbolism used, which we interpret as a rule-of-action "clarify symbolism". She then continued by repeating these steps when she identified similar instances of either unpacking a statement in her context or proving a similar Lean goal. We coded this as "repeat code sequence". Also, during coding, Gemma found an unexpected response from Lean when she wrote *apply pq* (Figures 2).



Figures 2: Screenshots from the Natural Number Game Level 4 - Gemma

Gemma: And then again, so I have the goal of  $q$  and I have the statement that  $p$  implies  $q$ , so I'm going to *apply pq*. No, I'm not. I'm working towards a random goal. Sorry. I have the [...] I want to work towards  $r$  because that's what I have. So I will have the goal of  $p$  [...] So I need to *apply qp* and then I have the goal of  $q$  and I have  $r$ . So I want to *apply rq* instead of *pq*.

We interpret this as a rule-of-action "check Lean goals and adapt code". Finally, once she has completed her proof, she reflected on the length of her proof:

Gemma: I was very hasty with splitting up the statement, say like  $p$  implies  $q$  and I had to prove that  $p$  implies  $q$ . Then I could have just done *exact* that statement instead of splitting it up all the way. [...] I just kind of saw it and thought oh I know how to split all of this up. So maybe just thinking about whether before I split everything up into its

individual parts, whether actually some bigger statements could be proven if I already have them.

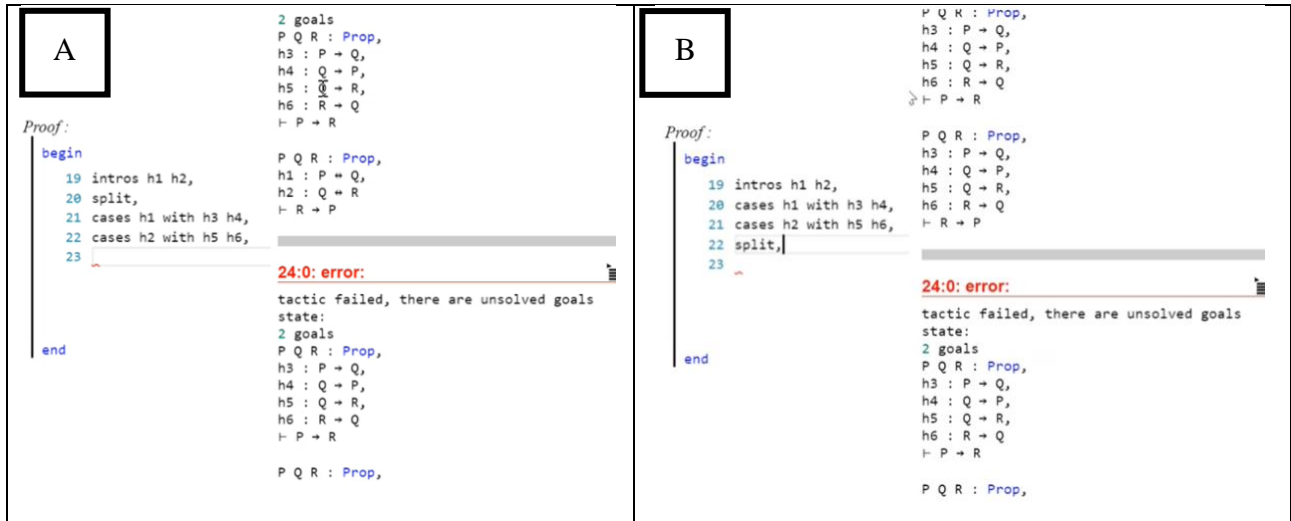
The rules-of-action we observed in Gemma’s activity are: “explore what is given”, “clarify symbolism”, “repeat code sequence” and “check Lean goals and adapt code”. We consider that the scheme she has developed in using Lean to write a proof of the transitive property is “verify step-by-step”.

We will now turn to describe the rules-of-action in Matt’s activity. He stated that although the instructions are saying that the *intro* tactic should be used, he will use *intros* instead. We interpret this as a rule-of-action “use a tactic which actions multiple steps”. Additionally, he introduces clear symbolism from the beginning in terms of the two hypotheses  $h_1$  and  $h_2$  which we consider as a rule-of-action “introduce clarifying symbolism”. He then realised that the *cases* tactic he used did not provide him with the expected result, which was to apply *split* to both Lean goals. In response to this feedback from Lean, he went back to his code to explore the order of tactics.

Matt: Hang on. So, when I [...] split it here. The *cases* is only applied to this first goal. So I wonder if [...] I were to just move that down here [...] that would carry to all the context. Yes, it would. Lovely.

This is a rule-of-action similar to Gemma’s “check Lean goals and adapt code” but is considering specifically the order of the tactics which we interpret as “move code for further efficiency” (Figures 3). Finally, considering what he had to prove after re-arranging his *split* and *cases* tactics he shares:

Matt: So we’ve got to prove  $p$  implies  $r$  from this. We’ve got  $p$  implies  $q$  and  $q$  implies  $r$ , we’ve got  $p$  implies  $r$  and  $r$  implies  $p$  so I’m wondering if we could just use the  $cc^2$  tactic and we want to use it twice. So, I wonder if we could do *repeat cc*. Yes, lovely. That all checks out or does what I thought it would do.



Figures 3: Screenshots from the Natural Number Game Level 4 - Matt

<sup>2</sup> Congruence closure is an operation in ITPs which determines whether the terms of a proof are equal under substitutivity of equality given a set of equalities.

We interpret what Matt did as a rule-of-action “use a tactic which actions multiple steps” as he warrants the use of *cc* but also considers that instead of writing it twice, he could use the *cc* tactic nested in the *repeat* tactic.

The rules-of-action we observed in Matt’s activity are: “use a tactic which actions multiple steps” “introduce clarifying symbolism”, “move code for further efficiency”. It is important to note here that Matt also had the subgoal to “write an efficient and quick proof”. We consider that the scheme he developed in using Lean to prove this statement is “verify multiple steps”.

<pre> Proof: begin   19 intro h,   20 cases h with pq qp,   21 intro j,   22 cases j with qr rq,   23 split,   24 intro p,   25 apply qr,   26 apply pq,   27 exact p,   28 intro r,   29 apply qp,   30 apply rq,   31 exact r, end </pre>	<pre> Proof: begin   19 intros h1 h2,   20 cases h1 with h3 h4,   21 cases h2 with h5 h6,   22 split,   23 repeat {cc}, end </pre>
Gemma’s complete proof	Matt’s complete proof

**Figures 4: Screenshots from the Natural Number Game Level 4 proofs by Gemma and Matt**

Figures 4 shows the complete proofs by Gemma and Matt. We note that there are differences between the two proofs in terms of their length and their use of tactics (e.g., *intro*, *cases \_\_\_ with \_\_\_*, *split*, *apply*, *exact*, *repeat* and *cc*). Both students introduced the relevant hypotheses and used the *split* and *cases* tactic. However, Matt (Figure 4, right) used the congruence closure (*cc*) tactic to conclude the task while Gemma did not. The use of this congruence closure tactic was discussed by both students during their interviews.

Gemma: I would rather write it out because that’s kind of the point of me doing it and it just feels like a proper proof. [...] *cc* kind of gets rid of it very quickly without me actually proving much. So... But for the point of like these worlds, I would prefer to actually write out what I’m doing and what the method is.

While Matt had a different approach to using *cc*.

Matt: *cc* is nice, it’s just if you’ve got, I guess it means if you’ve got like a path of implications that you have to sort of follow, *cc* just closes them.

Gemma therefore seems to avoid using the *cc* tactic aiming to see and check all the steps of her proof, which is in line with the earlier discussed scheme “verify step-by-step”. Whereas Matt tended to use *the cc* tactic often in his proof activity during the different levels of the Natural Number Game. Our analysis of their activity at other levels shows similar findings, signaling that these students may have developed two different instruments.

In their final interviews Matt and Gemma were asked to reflect on the experience of using Lean.

Matt: It’s like in Lean you’ve gotta look at what theorems you’ve got and how those could apply to, you know, your goal and your other hypotheses. And I think it just sort of builds

up a good way of thinking when it comes to just proving normal things [...] While I feel like I've gotten sort of an appreciation of like a method of problem solving from Lean. [...].

Matt considers the links between Lean and problem solving but shares that he is not sure whether he would continue to use Lean. He liked the way that Lean provides clearly what is available for the proof, the hypotheses, and the Lean goal. However, he feels that Lean can only provide him the problem-solving method but nothing further. On the other hand Gemma shared:

Gemma: Logically the steps kind of break it down because it's always the same and it's like a language [...] It's going to mirror what you're looking at in lectures, and it just kind of it's another way. [...] And I would just say it's a really good tool to check whether you're right or not as well. If you had a handwritten proof and you wanted to, you know, check how good your handwritten proof was, is, you can put it in and see whether it approves that as a proof kind of thing.

Gemma considered the links between the Lean proofs with the pen and paper proofs, discussed how Lean offers a different view at mathematics and noted the potential of the proof checking ability of Lean. Finally, she concluded that she would like to continue using Lean.

## Conclusion

This paper reports the schemes that two students – Matt and Gemma – developed when writing a proof for the same mathematical task using an interactive theorem prover in the context of the Natural Number Game. These schemes illustrate the different instruments they developed. Gemma seemed to use Lean as she would use pen and paper, by writing proofs that follow a more traditional layout without using *tactics* which solve multiple steps of the proof. Matt on the other hand uses features inherent to Lean (e.g., the *tactics*) to solve multiple proof goals at the same time, a proof strategy that is separate from proving with pen and paper. Therefore, the students seem to have developed two different schemes while interacting with the same artifact, namely: “verify step by step” and “verify multiple steps”. This observation is of interest to those – like us – who study the interaction of students with the programming language. The students in our study had the same experiences both in terms of teaching (they were in the same university cohort) and in terms of using Lean for the Natural Number Game and yet they exhibited different rules-of-action and developed different schemes when using the artefact. These considerations are important when introducing ITPs and encourage the researchers to think about what is that is learned through this experience and what are the factors that influence the difference in students' activity when using ITPs.

In the next steps of our work, we aim to analyse the schemes used by the four students while writing proofs for the other levels of the Natural Number Game. We are also interested in further exploring potential links between students' epistemologies and their scheme development. Finally, the Natural Number Game provides some extra features regarding the *tactics* and the *theorem statements*, it would be interesting to further examine students' activity using Lean without this extra support and further exploring aspects of the instrumentation.

## References

Avigad, J. (2019). Learning Logic and Proof with an Interactive Theorem Prover. In G. Hanna, D. Reid & M. de Villiers (Eds), *Proof Technology in Mathematics Research and Teaching*.



*Mathematics Education in the Digital Era* (vol 14, pp. 277–290). Springer.  
[https://doi.org/10.1007/978-3-030-28483-1\\_13](https://doi.org/10.1007/978-3-030-28483-1_13)

Bartzia, E., Meyer, A., & Narboux, J. (2022). Proof assistants for undergraduate mathematics and computer science education: elements of a priori analysis. In M. Trigueros, B. Barquero, R. Hochmut, & J. Peters (Eds.) *Proceedings of the Fourth Conference of the International Network for Didactic Research in University Mathematics (INDRUM)*. <https://hal.science/hal-03648357>

Bruijn, de N. G. (1980). A survey of the project Automath. In J. P. Seldin & J. R. Hindley (Eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, (p. 579–606). Academic Press Inc.

Gueudet, G., Buteau, C., Mesa, V., & Misfeldt, M. (2014). Instrumental and documentational approaches: From technology use to documentation systems in university mathematics education. *Research in Mathematics Education*, 16(2), 139–155.  
<https://doi.org/10.1080/14794802.2014.918349>

Gueudet, G., Buteau, C., Muller, E., Mgombelo, J., Sacristán, A. I., & Rodriguez, M. S. (2022). Development and evolution of instrumented schemes: a case study of learning programming for mathematical investigations. *Educational Studies in Mathematics*, 110(2), 353–377.  
<https://doi.org/10.1007/s10649-021-10133-1>

Hanna, G., Larvor, B. P., & Yan, X. K. (2023). Human-Machine Collaboration in the Teaching of Proof. *Journal of Humanistic Mathematics*, 13(1), 99–117.  
<https://scholarship.claremont.edu/jhm/vol13/iss1/7>

Iannone, P., & Simpson, A. (2022). How we assess mathematics degrees: the summative assessment diet a decade on. *Teaching Mathematics and its Applications: An International Journal of the IMA*, 41(1), 22–31. <https://doi.org/10.1093/teamat/hrab007>

Iannone, P., & Thoma, A. (2023). Interactive theorem provers for university mathematics: an exploratory study of students' perceptions. *International Journal of Mathematical Education in Science and Technology*. <https://doi.org/10.1080/0020739X.2023.2178981>

Lockwood, E., & Mørken, K. (2021). A Call for Research that Explores Relationships between Computing and Mathematical Thinking and Activity in RUME. *International Journal of Research in Undergraduate Mathematics Education*, 7(3), 404–416. <https://doi.org/10.1007/s40753-020-00129-2>

Sangwin, C. J., & O'Toole, C. (2017). Computer programming in the UK undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), 1133–1152. <https://doi.org/10.1080/0020739X.2017.1315186>

Thoma, A. & Iannone, P. (2022). Learning about proof with the theorem prover Lean: the abundant numbers task. *International Journal of Research in Undergraduate Mathematics Education*, 8(1), 64–93. <https://doi.org/10.1007/s40753-021-00140-1>

Vergnaud, G. (2009). The theory of conceptual fields. *Human development*, 52(2), 83–94.  
<https://doi.org/10.1159/000202727>